

ATVA 2008

International Symposium on Automated Technology for Verification and Analysis

October 20-23, 2008

Korea University, Seoul Korea

Day 1: October 20 (Monday)

09 : 00 - 10 : 00	Registration
10 : 00 - 12 : 00	Tutorial 1 : Logic in Specification and Verification (abstract) <i>Natarajan Shankar (SRI)</i> Session Chair : Sungdeok Cha
12 : 00 - 13 : 00	Lunch
13 : 00 - 15 : 00	Tutorial 2 : Boolean Modeling of Cell Biology (abstract) <i>David Dill (Stanford)</i> Session Chair : Dolon Peled
15 : 00 - 15 : 30	Coffee Break
15 : 30 - 17 : 30	Tutorial 3 : Checking Object Invariants by Combining Static and Dynamic Analysis (abstract) <i>Sriram Rajamani (Microsoft Research India)</i> Session Chair : Moonzoo Kim
17 : 30 - 20 : 00	Reception

Day 2: October 21 (Tuesday)

08 : 45 - 09 : 00	Opening
09 : 00 - 10 : 00	Keynote 1 : Tests, Proofs and Refinements <i>Sriram Rajamani (Microsoft Research India)</i>
10 : 00 - 10 : 30	Coffee Break
10 : 30 - 13 : 00	<p>Session 1 : Model Checking Session Chair : Yunja Choi</p> <p>CTL Model-Checking with Graded Quantifiers <i>Alessandro Ferrante, Margherita Napoli, Mimmo Parente</i></p> <p>Genetic Programming and Model Checking: Synthesizing New Mutual Exclusion Algorithms <i>Gal Katz, Doron Peled</i></p> <p>Computation Tree Regular Logic for Genetic Regulatory Networks <i>Radu Mateescu, Pedro T. Monteiro, Estelle Dumas, Hidde de Jong</i></p> <p>Compositional Verification for Component-based Systems and Application <i>Saddek Bensalem, Marius Bozga, Joseph Sifakis, Thanh-Hung Nguyen</i></p> <p>A Direct Algorithm for Multi-Valued Bounded Model Checking <i>Jefferson Andrade, Yukiyoishi Kameyama</i></p>
13 : 00 - 14 : 00	Lunch
14 : 00 - 15 : 50	<p>Session 2 : Software Verification Session Chair : Bow-Yaw Wang</p> <p>Model Checking Recursive Programs with Exact Predicate Abstraction <i>Arie Gurfinkel, Ou Wei, Marsha Chechik</i></p>

	<p>Loop Summarization using Abstract Transformers <i>Daniel Kroening, Natasha Sharygina, Stefano Tonetta, Aliaksei Tsitovich, Christoph Wintersteiger</i></p> <p>Dynamic Model Checking with Property Driven Pruning to Detect Race Conditions <i>Chao Wang, Yu Yang, Aarti Gupta, Ganesh Gopalakrishnan</i></p> <p>Authentication Revisited : Flaw or Not, the Recursive Authentication Protocol(short) <i>Guoqiang Li, Mizuhito Ogawa</i></p>
15 : 50 - 16 : 00	Coffee Break
16 : 00 - 17 : 00	Local Keynote : Software Quality in Consumer Electronics: Issues and Challenges <i>In-Kyung Ryu (LG Electronics Software Center)</i>
17 : 00 - 19 : 00	<p>Session 3 : Decision Procedures Session Chair : Mizuhito Ogawa</p> <p>Automating Algebraic Specifications of Non-freely Generated Data Types <i>Andriy Dunets, Gerhard Schellhorn, Wolfgang Reif</i></p> <p>Interpolants for linear arithmetic in SMT <i>Yuefeng Tang, Christopher Lynch</i></p> <p>SAT Modulo ODE: A Direct SAT Approach to Hybrid Systems <i>Andreas Eggers, Martin Frazle, Christian Herde</i></p> <p>SMELS: Satisfiability Modulo Equality with Lazy Superposition <i>Christopher Lynch, Duc-Khanh Tran</i></p>

Day 3: October 22 (Wednesday)

09 : 00 - 10 : 00	Keynote 2 : Formal Verification and Biology <i>David Dill (Stanford)</i>
10 : 00 - 10 : 30	Coffee Break
10 : 30 - 12 : 00	<p>Session 4 : Linear Time Analysis Session Chair : Saddek Bensalem</p> <p>Controllable Test Cases for the Distributed Test Architecture <i>Rob Hierons, Mercedes Merayo, Manuel Nunez</i></p> <p>Impartial Anticipation in Runtime-Verification (short) <i>Wei Dong, Martin Leucker, Christian Schallhart</i></p> <p>Run-time Monitoring of Electronic Contracts (short) <i>Marcel Kyas, Cristian Prisacariu, Gerardo Schneider</i></p> <p>Practical Efficient Modular Linear-Time Model-Checking (short) <i>Carlo Alberto Furia, Paola Spoletini</i></p>
12 : 00 - 13 : 00	Lunch
13 : 00 - 18 : 00	Excursion (Kyoung-Bok Palace + Korean Folk Museum/ Biwon)
18 : 00 - 20 : 00	Banquet

Day 4: October 23 (Thursday)

09 : 00 - 10 : 00	Keynote 3 : Trust and Automation in Verification Tools <i>Natarajan Shankar (SRI)</i>
10 : 00 - 10 : 15	Coffee Break

10 : 15 - 12 : 00	<p>Session 5 : Tool Demonstration Papers Session Chair : Martin Leucker</p> <p>Goanna: Syntactic Software Model Checking <i>Ralf Huuck, Ansgar Fehnker, Sean Seefried, Joerg Brauer</i></p> <p>A Dynamic Assertion-based Verification Platform for Validation of UML designs <i>Ansuman Banerjee, Sayak Ray, Pallab Dasgupta, Partha Pratim Chakrabarti, S. Ramesh, P Vignesh Ganesan</i></p> <p>CheckSpec: A Tool for Consistency and Coverage analysis of Assertion Specifications <i>Ansuman Banerjee, Kausik Datta, Pallab Dasgupta</i></p> <p>DiVinE Multi-Core - A Parallel LTL Model-Checker <i>Jiri Barnat, Lubos Brim, Petr Rockai</i></p> <p>Alaska: Antichains of Logic, Automata and Symbolic Kripke structures Analysis <i>Martin De Wulf, Laurent Doyen, Jean-Francois Raskin, Nicolas Maquet</i></p> <p>NetQi: A Model checker for Anticipation Game <i>Elie Bursztein</i></p> <p>Component-Based Design and Analysis of Embedded Systems with UPPAAL PORT <i>John Hakansson, Jan Carlson, Aurelien Monot, Paul Pettersson, Davor Slutej</i></p>	
12 : 00 - 13 : 30	Lunch	
13 : 30 - 15 : 50	<p>Session 6 : Timed & Stochastic Systems Session Chair : Gihwon Kwon</p> <p>Time-Progress Evaluation for Dense-Time Automata with Concave Path Conditions <i>Farn Wang</i></p> <p>Decidable Compositions of O-minimal Automata <i>Alberto Casagrande, Pietro Corvaja, Carla Piazza, Bud Mishra</i></p> <p>On the Applicability of Stochastic Petri Nets for Analysis of Multiserver Retrial Systems with Different Vacation Policies <i>Nawel Gharbi</i></p> <p>Model Based Importance Analysis for Minimal Cut Sets <i>Eckard Bode, Thomas Peikenkamp, Jan-Hendrik Rakow, Samuel Wischmeyer</i></p> <p>Passive Testing of Timed Systems(short) <i>Cesar Andres, Mercedes Merayo, Manuel Nunez</i></p>	Tool Demonstrations
15 : 50 - 16 : 30	Coffee Break	
16 : 30 - 18 : 30	<p>Session 7 : Theory Session Chair : Manuel Nunez</p> <p>Approximate Invariant Property Checking Using Term-Height Reduction for a Subset of First-Order Logic <i>Hiroaki Shimizu, Kiyoharu Hamaguchi, Toshinobu Kashiwabara</i></p> <p>Tree Pattern Rewriting Systems <i>Blaize Genest, Anca Muscholl, Olivier Serre, Marc Zeitoun</i></p> <p>Deciding Bisimilarity of Full BPA Processes Locally <i>Lingyun Luo</i></p> <p>Optimal Strategy Synthesis in Request-Response Games <i>Florian Horn, Wolfgang Thomas, Nico Wallmeier</i></p>	

Tutorial 1

Title	Logic in Specification and Verification
Speaker	Natarajan Shankar (SRI)
Abstract	We explore the effective use of logic with automated tools such as PVS, Yices, and SAL. We look at the range of choices in logics and languages, modeling and formalization, and automation and interaction. PVS is a general-purpose framework for specification and verification. SAL is specialized to transition systems. Yices is a solver for satisfiability modulo theories.

Tutorial 2

Title	Boolean Modeling of Cell Biology
Speaker	David Dill (Stanford)
Abstract	Over the years, models similar to digital logic have been used in cell biology. Because much of the initial work was done by people not from the digital design community, there has been a certain amount of parallel evolution, where the questions asked and techniques used are sometimes different from the approaches used in the design of digital systems designed by engineers. This tutorial will discuss why Boolean models of cell biology might be appropriate and useful, survey past work in the area, and compare with approaches used in analysis of human-designed systems, such as model checking.

Tutorial 3

Title	Checking Object Invariants by Combining Static and Dynamic Analysis
Speaker	Sriram Rajamani (Microsoft Research India)
Abstract	<p>We consider object protocols that constrain interactions between objects in a program. The goal of an object protocol is expressed as a protocol invariant. Fundamental properties such as ownership can be expressed as protocol invariants. We present a language, PROLANG, to specify object protocols along with their protocol invariants, and a tool, INVCOP++, to check if a program satisfies a protocol invariant. INVCOP++ separates the problem of checking if a protocol satisfies its protocol invariant (called protocol correctness), from the problem of checking if a program conforms to a protocol (called program conformance).</p> <p>The former is solved using static analysis, and the latter using runtime analysis. We present theoretical guarantees about the way we combine static and runtime analysis, and empirical evidence that our tool INVCOP++ finds usage errors in widely used APIs. We also show that statically checking protocol correctness greatly optimizes the overhead of checking program conformance, thus enabling API clients to test whether their programs use the API as intended by the API designer.</p>